

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

3/5/1 (Item 1 from file: 351)
DIALOG(R) File '351:Derwent WPI
(c) 2003 Thomson Derwent. All rts. reserv.

012771392 **Image available**
WPI Acc No: 1999-577615/ 199949
XRPX Acc No: N99-426772

Failure cause indication system of computer implemented failure management system for communication network - has user interface which indicates fundamental cause of network failure, specified by failure management unit, to user

Patent Assignee: SUMITOMO ELECTRIC IND. CO (SUME)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
JP 11252074	A	19990917	JP 97134967	A	19970526	199949 B

Priority Applications (No Type Date): JP 97134967 A 19970526

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 11252074	A		14	H04L-012/24	

Abstract (Basic): JP 11252074 A

NOVELTY - A failure management unit (34) specifies fundamental cause of network failure based on convergence of symptom event and symptom event pattern matrix. User interface (36) indicates the fundamental cause of the network failure, specified by the failure management unit to user. DETAILED DESCRIPTION - A component management unit (30) is connected to a network interface (24) which acquires and outputs component information data and failure information data of a management objective network. The component management unit performs build-up of management objective object component information and symptom event pattern matrix from component information data output from network interface. The component information data includes management objective object model and event propagation model. An event database (26) connected with the network interface and component management unit, maintains an event corresponding to component information data and failure information data of the network, from the network interface. A failure management unit connected with the component management unit and event data base, receives convergence of symptom event and symptom event pattern matrix from the component management unit.

USE - In failure management system for communication network.

ADVANTAGE - Enables to specify cause of failure of network correctly even when all symptom events are not observed correctly.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of network management system. (24) Network interface; (26) Event database; (30) Component management unit; (34) Failure management unit; (36) User interface.

Dwg.1/18

Title Terms: FAIL; CAUSE; INDICATE; SYSTEM; COMPUTER; IMPLEMENT; FAIL; MANAGEMENT; SYSTEM; COMMUNICATE; NETWORK; USER; INTERFACE; INDICATE; FUNDAMENTAL; CAUSE; NETWORK; FAIL; SPECIFIED; FAIL; MANAGEMENT; UNIT; USER

Derwent Class: W01

International Patent Class (Main): H04L-012/24

International Patent Class (Additional): H04L-012/26; H04L-029/14

File Segment: EPI

3/5/2 (Item 1 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2003 JPO & JAPIO. All rts. reserv.

06310476 **Image available**

NETWORK MANAGEMENT SYSTEM

PUB. NO.: 11-252074 A]
PUBLISHED: September 17, 1999 (19990917)
INVENTOR(s): TAKENAMI YOSHINORI
KUWABARA NORIAKI
APPLICANT(s): SUMITOMO ELECTRIC IND LTD
APPL. NO.: 09-134967 [JP 97134967]
FILED: May 26, 1997 (19970526)
INTL CLASS: H04L-012/24; H04L-012/26; H04L-029/14

ABSTRACT

PROBLEM TO BE SOLVED: To provide an NMS(network management system) which can accurately specify the cause of a network fault even through all symptom events cannot be accurately observed.

SOLUTION: This NMS is provided with a network interface part and a constitution management part which holds a management object model and an event propagation model and constitutes a symptom event pattern matrix. The management object model includes the relation 70 expressing the azimuth relation on the network connection to a network management system that is defined in relation to the connecting relation between the objects. The event propagation model is prepared taking into consideration of the relation 70, expressing the azimuth relation as well and the symptom event pattern matrix is prepared so as not to include the symptoms which cannot be observed in the NMS.

COPYRIGHT: (C)1999,JPO

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-252074

(43)公開日 平成11年(1999) 9月17日

(51)Int.Cl.⁹

識別記号

F I

H 0 4 L 12/24
12/26
29/14

H 0 4 L 11/08
13/00

3 1 3

審査請求 未請求 請求項の数 3 O L (全 14 頁)

(21)出願番号 特願平9-134967

(22)出願日 平成 9 年(1997) 5 月26日

特許法第64条第2項ただし書の規定により図面第13図の一部は不掲載とした。

(71)出願人 000002130

住友電気工業株式会社

大阪府大阪市中央区北浜四丁目 5 番33号

(72)発明者 武並 佳則

大阪市此花区島屋一丁目 1 番 3 号 住友電
気工業株式会社大阪製作所内

(72)発明者 桑原 教彰

大阪市此花区島屋一丁目 1 番 3 号 住友電
気工業株式会社大阪製作所内

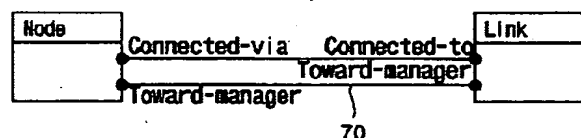
(74)代理人 弁理士 深見 久郎 (外 2 名)

(54)【発明の名称】 ネットワーク管理システム

(57)【要約】

【課題】 すべての症状イベントを正確に観測することができなくとも、正確にネットワークの障害の原因を特定することができるネットワーク管理システムを提供する。

【解決手段】 ネットワーク管理システムは、ネットワークインタフェース部と、管理対象オブジェクトモデルとイベント伝播モデルとを保持し症状イベントパターンマトリックスを構成する構成管理部とを含む。管理対象オブジェクトモデルは、オブジェクト間の接続関係に関連して定義された、ネットワーク管理システムに対するネットワーク接続上での方位関係を表わす関係を含んでいる。イベント伝播モデルは、方位関係を表わす関係 70 をも考慮して予め準備されており、症状イベントパターンマトリックスは、NMS で観測できない症状を含まないように予め準備される。



【特許請求の範囲】

【請求項1】 管理対象ネットワークの構成情報データおよび障害情報データを取得し出力するネットワークインターフェイス部と、

前記ネットワークインターフェイス部に接続され、管理対象オブジェクトモデルとイベント伝播モデルとを保持し、前記ネットワークインターフェイスから与えられる前記ネットワークの前記構成情報データから管理対象オブジェクト構成情報および症状イベントパターンマトリックスを構築するための構成管理部とを含むネットワーク管理システムであって、

前記管理対象オブジェクトモデルは、オブジェクト間の接続関係に関連して定義された、ネットワーク管理システムに対する、オブジェクト間のネットワーク接続上での方位関係を含み、

前記イベント伝播モデルは、前記方位関係をも考慮して予め準備されており、

前記ネットワーク管理システムはさらに、

前記構成管理部に接続された管理対象オブジェクトリポジトリと、

前記ネットワークインターフェイス部に接続され、前記ネットワーク上の構成情報データと障害情報データとを前記ネットワークインターフェイス部からイベントとして受けて保持するためのイベントデータベース部と、

前記構成管理部および前記イベントデータベース部に接続され、前記構成管理部から症状イベントパターンマトリックスを、前記イベントデータベース部から症状イベントの集合を受け、前記症状イベントパターンマトリックスおよび前記症状イベントの集合に基づいて前記ネットワーク障害の根本原因の問題を特定するための障害管理部と、

前記障害管理部が特定した前記障害の根本原因の問題をユーザに提示するためのユーザインターフェイス部とを含む、ネットワーク管理システム。

【請求項2】 前記構成管理部は、前記管理対象オブジェクトモデルと前記イベント伝播モデルと、前記ネットワークインターフェイス部から与えられる実際のネットワーク上の構成情報データとに基づき、前記ネットワーク管理システムの位置を考慮にいれて管理対象オブジェクト間の関係を自動的に設定する、請求項1に記載のネットワーク管理システム。

【請求項3】 前記構成管理部は、前記ユーザインターフェイス部を介して前記ネットワークの接続関係の変更情報が入力されたことに応答して、管理対象オブジェクト間の関係を自動的に設定する、請求項2に記載のネットワーク管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ネットワーク上の障害を管理するネットワーク管理システムに関し、特

に、ネットワーク上で観測されるさまざまな複数の障害の症状から障害の根本原因を特定する機能を有するネットワーク管理システムに関する。

【0002】

【従来の技術】コンピュータによる通信ネットワークの大規模化が進んでいる。通信ネットワークが大規模化するに従って、ネットワーク上に発生する障害の及ぼす影響も大規模かつ深刻なものとなりつつある。そのためネットワーク管理をいかに効率よく行なうか、が非常に重要である。以下、本明細書上で使用されるネットワーク管理に関する用語について定義をする。

【0003】「イベント」とは、ネットワークにおいて発生する例外的な状態のことをいう。ハードウェアやソフトウェアの故障、停止、性能のボトルネック、ネットワークの構成の不整合、設計不十分による意図せざる結果、コンピュータウィルス等の悪意による被害などを含む。「不具合」は「イベント」と同じ意味で使用する。

【0004】「症状」とは、観測可能なイベントのことをいう。「症状イベント」と同じである。たとえば「ある宛先Aに対して常に通信に時間がかかり再送信が必要となる」、「ある宛先Bに対していつも文字化けが生ずる」、「ある宛先Cに対していつも受信確認が返ってこない」などの事象をいう。「問題」とは、障害の根本原因のことをいう。必ずしも観測可能ではない。たとえば通信装置の送信機破損、通信ケーブルの断線、通信回線の容量不足などが例である。「問題イベント」は「問題」と同じ意味である。

【0005】「オブジェクト」とは、概念や抽象または対象となる問題に対して明確な境界と意味とを持つ何物か、のことをいう。「オブジェクトインスタンス」とは、ある特定の1つのオブジェクトのことをいう。「オブジェクトクラス」とは、同様の性質（属性）、共通の振る舞い（操作）、他のオブジェクトとの共通の関係、および共通の意味を持つオブジェクトのグループをいう。「クラス」はオブジェクトクラスと同じである。

「サブクラス」とは、あるクラスの下位クラスとして、そのクラスに包含されるクラスのことをいう。「属性」とは、クラスに属する各オブジェクトによって保持されるデータをいう。

【0006】「オブジェクト図」とは、オブジェクト、クラス、それらの間の関係のモデル化のための、形式的な図式記法をいう。「クラス図」とは、多くの可能なインスタンスを記述するためのスキーマ、パターン、テンプレートなどである。

【0007】「リポジトリ」とは、必要な情報を集約して一覧表的な形式で記憶した記憶部を言う。集約一覧表メモリとでも呼ぶべきものである。「イベントリポジトリ」とは、実際に発生した症状パターンを記憶する記憶部をいう。「オブジェクトリポジトリ」とは、ネットワークの構成モデルのリポジトリのことをいう。これによ

って「ある原因を仮定したときに発生する症状パターン」を前もって特定しておくことができる。

【0008】ネットワークのあるリソースにおける1つの問題イベントは、関係する複数のリソースの多くの症状イベントを引き起こし得る。問題の中には、観測可能なイベントであるものもあるが、一般には必ずしも観測可能ではない。そのため複数の症状から障害の根本原因である問題を特定する必要がある。したがって、ネットワーク管理者は、根本原因の問題を特定するために、観測される種々の症状イベントを問題と関連させることができなければならない。

【0009】しかし、ネットワークが大規模になると、観測される症状イベントの数も膨大になる。またどの問題がどの症状を引き起こすかという「因果関係」とでも言うべきものも複雑になってくるために、オペレータが手作業で障害の根本原因の問題を特定することはほとんど不可能となる。

【0010】このようなネットワーク上で観測される膨大な障害の症状イベントから根本原因の問題を正確にかつ高速に特定するための従来技術手法として、1996年6月18日発行の米国特許第5,528,516号

(「Apparatus and Method for Event Correlation and Problem Reporting (イベント相関および問題報告装置および方法)」)が提案されている。

【0011】この従来技術は次の2つの技術に分けることができる。

- (1) 管理対象ネットワークのモデリング技術
- (2) イベント相関技術

「管理対象ネットワークのモデリング技術」とは、実際のネットワーク上で発生する問題イベントおよび症状イベントをいかに正確に効率よくモデル化するかという技術である。(2)のイベント相関技術とは、主として、観測される膨大な症状イベントからいかに高速に根本原因である問題を特定するか、という技術である。本発明は(1)の管理対象ネットワークのモデリング技術に関し、イベント相関技術には関連しない。そのため以下では(2)のイベント相関技術については詳細には説明しない。なお以下の説明は、障害に関するイベントに限定して行なうが、イベントの種類はどのようなものであってもよく、本発明は障害に関するイベントのみに制約されるものではない。

【0012】この従来技術で提案されている「管理対象ネットワークのモデリング技術」について以下に簡単に説明する。まず、ルータおよびハブなどのネットワーク機器と、パーソナルコンピュータおよびワークステーションなどのコンピュータと、これらを接続するネットワークそのものと、コンピュータ等の上で実行されるソフトウェア等とからなるすべてのものを管理対象オブジェクトManagedObjectとしてモデル化する。そして、オブジェクト間の「関係」を重要視し、ネットワーク上で発

生する問題イベントおよび症状イベントは、管理対象オブジェクト間に設定された関係に沿って「伝播」するものとしてイベントの伝播をモデル化する。

【0013】上述の従来技術では、このような管理対象オブジェクトのモデル化およびイベント伝播のモデル化の静的な側面を抽象化し、モデル化を効率的に行なうためにオブジェクト指向の概念を導入している。すなわち種々の管理対象オブジェクトをクラスとしてモデル化する。そしてクラス間の関係を定義する。さらにあるイベントは、クラス間の関係に沿って伝播するものとしてモデル化される。オブジェクト指向技術については種々の教科書があるのでそれらを参照されたい。

【0014】まず、管理対象オブジェクトクラスのモデル化について説明する。図12に、上述の従来技術で提案されている管理対象オブジェクトモデル(クラス階層)を示す。図12および他の図面において、管理対象オブジェクトモデルについてはOMT記法を用いて記す。図12を参照して、各ボックスはオブジェクトクラスを示す。オブジェクトクラスの各ボックスは2つのボックスに分割されている。上部のボックスに記載されているのはオブジェクトクラス名であり、下部のボックスに記載されているのはそのクラスのオブジェクトが持つべき属性である。各クラス間には関係が設定され、各関係はクラス間に引かれた線で示されている。各線の近傍には、その関係の名称が記載されている。

【0015】図12に示されるモデルは、ManagedObject クラス110と、Element クラス112と、Layer クラス114と、Group クラス116と、Resourceクラス118と、PhysicalService クラス120と、LogicalServiceクラス122と、Nodeクラス124と、Linkクラス126と、Manager クラス128とを含む。Manager クラス128がネットワーク管理システムのクラスである。

【0016】これらクラス間に設定される関係は、その関係により結びつけられる2つのクラスのいずれから見るかによって名称が変わる。たとえばManager クラス128とResourceクラス118との関係は、Manager クラス128から見ればManages、Resourceクラス118から見ればManaged-byとなる。またたとえばNodeクラス124とLinkクラス126との間の関係208は一方から見ればConnected-via、他方から見ればConnected-toである。またたとえばElement クラス112から出て同じクラス内に戻る関係206(Layered-overとUnderlying)のように、同一クラス内のオブジェクト間に定められる関係もある。

【0017】今、実際のネットワークの例として図13に示されるネットワークを考える。このネットワークは、互いにブリッジ140によって結合された2つのイーサネットワーク134および136を含む。一方のイーサネットワーク134にはホスト130が、他方のイーサネットワーク136にはホスト132がそれぞれ接続されているものとする。以下の例ではホスト130にネットワーク管理

システムが搭載されているものとして考える。

【0018】図13に示されるネットワークを図12に示されるような考え方をを用いてモデル化するために次のようにする。まず図14に示されるように、Nodeクラス124の下に3つのサブクラスTcpNode クラス154、IpNode クラス156 およびEtherNode クラス158 を定義する。さらにLinkクラス126 のサブクラスとしてTcpLink クラス160、IpLinkクラス162、EtherLink クラス164 およびMacBridgeLink クラス166 を定義する。なお、各クラスにはそのクラス特有の属性を定義する必要がある。たとえばIpNodeクラスにはIP(Internet Protocol) アドレス属性を定義するなど、である。しかし、本発明には属性の定義は関係しないので、ここでは属性についての詳細な説明を行なうことはしない。

【0019】図14に示されるクラスを用いると、図13に示されるネットワーク例は図15のように表現することができる。図15は、オブジェクト指向技術では「インスタンス図」と呼ばれる。

【0020】図15を参照して、ホスト130 には、それぞれManager クラス、TcpNode クラス、IpNodeクラスおよびEtherNode クラスのインスタンスであるインスタンス170、172、174 および176 が含まれる。一方ホスト132 には、TcpNode クラス、IpNodeクラスおよびEtherNode クラスのインスタンスであるインスタンス182、184 および186 が含まれる。インスタンス172 および182 はTcpLink クラスのインスタンス200 によって結合される。インスタンス174 および184 はIpLinkのインスタンスであるインスタンス202 によって結合される。

【0021】ブリッジ140 は、EtherNode クラス、MacBridgeNode クラスおよびEtherNodeクラスのインスタンスであるインスタンス190、192 および194 を含んでいる。インスタンス176 および190 はEtherLink クラスのインスタンスであるインスタンス196 により結合される。インスタンス194 および186 は同じくEtherLink クラスのインスタンス198 により結合される。インスタンス190、192、194、196 および198 はそれぞれインスタンス202 と結合されている。

【0022】このようにしてモデル化された管理対象オブジェクトに対して、症状イベントの伝播ルールが予め準備される。この伝播ルールは、障害の根本原因の問題イベントが障害の症状イベントに伝播し、その症状イベントが別の症状イベントに伝播するという関係をルール化したものである。この伝播ルールの集合を伝播モデルと呼ぶ。伝播モデルの例を以下に示す。

【0023】・EtherLink がDownすれば、Connected-to 関係に従ってEtherNode がDownする。

【0024】・EtherNode がDownすれば、Underlying関係に沿ってIpNodeがDownする。

・EtherNode がDownすれば、Connected-via 関係に沿ってMacBridgeLink がCannotBridgingである。

【0025】・IpNodeがDownすれば、Underlying関係に沿ってTcpNode がDisconnectする。

・TcpNode がDisconnectすれば、Connected-via 関係に沿ってTcpLink がNotResponding である。

【0026】・TcpLink がNotResponding であると、Connected-to関係に沿ってそれに接続されるTcpNode がDisconnectである。

【0027】上記したルールにおいて、Down、DisconnectおよびNotResponding が障害の問題イベントおよび/または症状イベントである。イベントの中には問題イベントでかつ症状イベントであるものもあるし、どちらでもないものもある。このような伝播モデル(ルール)を図式化して図16に示す。図16において、上記したイベントは、管理対象オブジェクトのクラス間に定義されている関係に沿って伝播するという、イベント伝播のモデル化がなされている。図16に示される各クラス154、156、158、160、164 および166 の間では、それぞれの関係210、212、214、216、218 および220 という関係(順不同)があり、上記したルールに従ってこれらの関係に沿ってイベントが伝播する。

【0028】図6に示されるイベント伝播モデルが想定されている場合に、図13に示され図15にモデル化されているネットワークでイベントがどのように伝播するかを以下に示す。ここでは障害の根本原因の問題としてEtherLink1がDown(これを「ケース1」と称する)とEtherLink2がDown(これを「ケース2」と称する)の場合を考える。この場合、イベント伝播モデルに従って観測されると記載される症状イベントは図17の表1に示されるようになる。

【0029】これらの表において、最も左側の列の各欄には症状イベントが、第1行目の各欄には問題イベントが、それぞれ記載されている。ある問題イベントの列とある症状イベントの行との交わる欄には、その問題イベント(原因)が発生したときに、当該症状イベントが発生するか否かを示す。この場合症状イベントが生ずる場合に「1」を記入するものとする。たとえばEtherLink1がDownが発生したときにはEcpNode1.Disconnect が生じるが、EtherNode2.Down は生じない。なお各インスタンス名は図15に記載のものをを用いる。各ケースについて該当の列を縦に見ていくと、「1」および「空白

(0)」を連ねたものが得られる。これを当該問題イベントに対する症状イベント群のコードと称する。たとえば表1のケース1では症状イベント群のコードは「1101101101」である。未記入(空白)の箇所は0としてある。

【0030】図17に示される表1からわかるように、問題イベントが異なると、観測される症状イベント群のコードが異なる。したがって実際にネットワーク管理システム(NMS)により観測および収集される症状イベント群と、図7の表1に示される症状イベント群とを比較

することにより、障害の根本原因の問題を特定することが可能となる。

【0031】

【発明が解決しようとする課題】このようにある問題イベントに対して生ずる症状イベントを予めパターン化しておけば、実際に障害が生じたときの症状パターンとこのパターンとを比較するという比較的単純な作業により障害の根本原因の問題を特定することができる。したがって、この従来の技術により障害の根本原因の問題の特定が非常に容易になるかと思われる。しかしこの従来の技術には次のような問題点がある。

【0032】たとえば既に述べたように、図3のホスト130上でNMSソフトウェアが起動されている場合を考える。「EtherLink1がDown」が生ずると、図15からも容易にわかるように、NMSは第2のイーサネット136側と通信できなくなる。したがってNMSは第2のイーサネット136側の症状イベントを観測することができない。そのためある障害が発生した場合に、実際にNMSが観測し収集する症状イベント群が、予め準備された症状イベントパターンマトリックスと一致しなくなる。

【0033】図17の表1に示されるケース1およびケース2の場合にそれぞれNMSにより実際に観測され得る症状イベントを図18の表2として示す。表1と表2とを比較すると容易にわかるように、同じ問題に対する伝播モデルでの症状イベントパターンマトリックスと、実際に得られる症状イベント群とは異なっている。より具体的には、ケース1（実際）とケース1（伝播モデル）との症状パターンの差は3であり、ケース1（実際）とケース2（伝播モデル）との症状パターンの差は7である。またケース2（実際）とケース2（伝播モデル）との症状パターンの差は2であり、ケース2（実際）とケース1（伝播モデル）との症状パターンの差は6である。このように実際に生じているはずの症状イベントがNMSに観測されない場合には、NMSが観測し収集する症状イベント群と、予め準備された症状イベントパターンマトリックスとはそれほど一致しない。症状イベントが観測されない場合だけでなく、偽の症状イベントが観測される場合においても同様である。

【0034】上記した従来の技術では、こうした場合でもある程度まで正確に原因を特定することが可能なロバストなアルゴリズムが提案されている。しかし、ネットワークが大規模化・複雑化した場合に、観測されないイベントの数が増えてくると、適切な原因特定処理ができなくなるおそれが非常に大きい。

【0035】上記目的を達成する場合にも、管理対象オブジェクトモデルの構築は容易でなければならない。また、ネットワークの接続関係に変更があるごとに管理対象オブジェクトモデルも変更しなければならないので、管理対象オブジェクトモデルの維持も容易に行なうことが必要である。

【0036】それゆえに請求項1に記載の発明の目的は、すべての症状イベントを正確に観測することができなくても、正確にネットワークの障害の原因を特定することができるネットワーク管理システムを提供することである。

【0037】請求項2に記載の発明の目的は、請求項1に記載の発明の目的に加え、管理対象オブジェクトモデルの構築が容易なネットワーク管理システムを提供することである。

10 【0038】請求項3に記載の発明の目的は、請求項2に記載の発明の目的に加え、ネットワークの接続関係に変更があるごとに容易に管理対象オブジェクトモデルの維持を行なうことができるネットワーク管理システムを提供することである。

【0039】

【課題を解決するための手段】請求項1に記載のネットワーク管理システムは、管理対象ネットワークの構成情報データおよび障害情報データを取得し出力するネットワークインターフェイス部と、ネットワークインターフェイス部に接続され、管理対象オブジェクトモデルとイベント伝播モデルとを保持し、ネットワークインターフェイスから与えられるネットワークの構成情報データから管理対象オブジェクト構成情報および症状イベントパターンマトリックスを構築するための構成管理部とを含む。管理対象オブジェクトモデルは、オブジェクト間の接続関係に関連して定義された、ネットワーク管理システムに対する、オブジェクト間のネットワーク接続上での方位関係を含んでいる。イベント伝播モデルは、方位関係をも考慮して予め準備されている。ネットワーク管理システムはさらに、構成管理部に接続された管理対象オブジェクトトリボジトリと、ネットワークインターフェイス部に接続され、ネットワーク上の構成情報データと障害情報データとをネットワークインターフェイス部からイベントとして受けて保持するためのイベントデータベース部と、構成管理部およびイベントデータベース部に接続され、構成管理部から症状イベントパターンマトリックスを、イベントデータベース部から症状イベントの集合を受け、症状イベントパターンマトリックスおよび症状イベントの集合に基づいてネットワーク障害の根本原因の問題を特定するための障害管理部と、障害管理部が特定した障害の根本原因の問題をユーザに提示するためのユーザインターフェイス部とを含む。

40 【0040】管理対象オブジェクトモデルが、ネットワークの接続上でのネットワーク管理システムとの方位関係を含んでいる。イベント伝播モデルは、単にオブジェクト間の接続関係だけでなく、ネットワーク管理システムとの方位関係をも考慮して準備されている。そのため症状イベントパターンマトリックスの作成時に、各障害が発生したとしてもネットワーク管理システムに伝播しないイベント、すなわちネットワーク管理システムに観

測されないイベントは考慮されない。実際の障害が発生したときにも、あるイベントはネットワーク管理システムに到達しないことがあるが、本発明のネットワーク管理システムではそのような症状イベントは、症状イベントパターンマトリックスの作成においては症状イベントとして考慮されない。そのため実際の症状イベント群と構成管理部により構築された症状イベントパターンマトリックスとはその内容がよく一致する。したがって、すべての症状マトリックスが正確に観測されなくても、正確にネットワークの障害の原因を特定することができる。

【0041】請求項2に記載の発明に係るネットワーク管理システムは、請求項1に記載のシステムであって、構成管理部は、管理対象オブジェクトモデルとイベント伝播モデルと、ネットワークインターフェイス部から与えられる実際のネットワーク上の構成情報データとに基づき、ネットワーク管理システムの位置を考慮にいれて管理対象オブジェクト間の関係を自動的に設定する。

【0042】管理対象オブジェクト間の関係が自動的に設定されるので、管理対象オブジェクトモデルの構築が容易である。

【0043】請求項3に記載の発明に係るネットワーク管理システムは、請求項2に記載のシステムであって、構成管理部は、ユーザインターフェイス部を介してネットワークの接続関係の変更情報が入力されたことに応答して、管理対象オブジェクト間の関係を自動的に設定する。

【0044】システムの接続関係に変更があると、管理対象オブジェクト構成情報も当然変更される。この変更によって当然に症状イベントパターンマトリックスも変更する必要があるが、この変更がユーザインターフェイスを用いて行なわれた変更情報の入力に応答して自動的に設定される。したがってネットワークの接続関係に変更があるごとに容易に管理対象オブジェクトモデルの維持を行なうことができる。

【0045】

【発明の実施の形態】実施の形態1

従来の技術で述べた問題点が生ずるのは、従来技術で提案された管理対象オブジェクトモデルおよびイベント伝播モデルでは、管理対象ネットワークと、NMSが管理するために使用するネットワーク（「管理ネットワーク」と称する）とが、モデルとして別であるような理想的な状態が仮定されているためであると考えられる。こうした理想的な状態では、すべての管理対象の症状がNMSにより観測可能となるので、予め準備された症状イベントパターンマトリックスと実際の症状イベントの集合とは一致するはずである。しかし実際のコンピュータネットワークでは、管理ネットワークは管理対象ネットワークと同一である場合が多い。こうした場合には既に述べたようにすべての症状がNMSにより観測可能であるわけで

はなく、そのため今回のような問題が生ずる。以下に述べる実施の形態では、上記問題を解決するために、管理対象オブジェクトモデルをNMSと管理対象オブジェクト群との位置関係をも考慮したモデルとしている。

【0046】図1に本願発明に係るネットワーク管理システムの第1の実施の形態のシステム20をブロック図形式で示す。図1を参照してネットワーク管理システム20は、管理対象ネットワーク22に接続され、SNMP (Simple Network Management Protocol) 等のネットワーク管理プロトコルを用いて管理対象装置のMIB (Management Information Base) 等のデータを取得したりポーリングしたりするためのネットワークインターフェイス部24と、ネットワークインターフェイス部24からネットワークの構成情報に関するデータを受け、後述するように改善された管理対象オブジェクトモデルと、これに関連した改善後のイベント伝播モデルとを保持するとともに、これらモデルとネットワークの構成情報とに基づき管理対象オブジェクト構成情報を構築するための構成管理部30と、構成管理部30により構築された管理対象オブジェクト構成情報を保持するための管理対象オブジェクトトリポジトリ32と、構成情報データおよび障害情報データをイベントとして保持するためのイベントトリポジトリ28と、ネットワークインターフェイス部24から構成情報データおよび障害情報データをイベントとして受取り、イベントトリポジトリ28に保持させるためのイベントデータベース部26と、構成管理部30から症状イベントパターンマトリックスを受取って保持し、イベントデータベース部26から障害の症状イベントの集合を受け、症状イベントパターンマトリックスの症状イベントパターンと、実際に観測される症状イベントパターンとを比較して、障害の根本原因の問題を特定するための障害管理部34と、障害管理部34から障害の根本原因の問題を特定する情報を受け、ユーザに提示するためのユーザインターフェイス部36とを含む。

【0047】本実施の形態では、管理対象オブジェクトモデルとイベント伝播モデルとに改善を行なっているが、それらについては図4以降を参照して後述することとする。

【0048】図1に示されるネットワーク管理システムは、実際にはパーソナルコンピュータまたはワークステーションなど、コンピュータ上で実行されるソフトウェアにより実現される。図2に、ネットワーク管理システムを実現するコンピュータの外観を示す。図2を参照してこのコンピュータは、CD-ROM (Compact Disc Read-Only Memory) ドライブ50およびFD (Flexible Disk) ドライブ52を備えたコンピュータ本体40と、ディスプレイ42と、プリンタ44と、キーボード46と、マウス48とを含む。

【0049】図3に、このコンピュータの構成をブロック図形式で示す。図3に示されるようにこのシステム20

を構成するコンピュータの本体40は、CD-ROMドライブ50およびFDドライブ52に加えて、それぞれバス66に接続されたCPU (Central Processing Unit) 56と、ROM (Read Only Memory) 58と、RAM (Random Access Memory) 60と、ハードディスク54とを含んでいる。CD-ROMドライブ50にはCD-ROM62が装着される。FDドライブ52にはFD64が装着される。

【0050】既に述べたようにこのネットワーク管理システムは、コンピュータハードウェアと、CPU 56により実行されるソフトウェアとにより実現される。一般的にこうしたソフトウェアは、CD-ROM62、FD64などの記憶媒体に格納されて流通し、CD-ROMドライブ50またはFDドライブ52などにより記憶媒体から読取られてハードディスク54に一旦格納される。さらにハードディスク54からRAM 60に読出されてCPU56により実行される。図2および図3に示したコンピュータのハードウェア自体は一般的なものである。したがって、本発明の最も本質的な部分はCD-ROM62、FD64、ハードディスク54などの記憶媒体に記憶されたソフトウェアである。

【0051】なお図2および図3に示したコンピュータ自体の動作は周知であるので、ここではその詳細な説明は繰返さない。

【0052】以下、図1～図3に示す本願発明の実施の形態に係るネットワーク管理システム20の動作について説明する。ネットワークインターフェイス部24は、SNMP等のネットワーク管理プロトコルを用いて、管理対象装置のMIB等のデータを取得したりポーリングしたりする。またネットワークインターフェイス部24は、管理対象装置からのトラップイベントを受信する。ネットワークインターフェイス部24はさらに、管理対象装置から取得したデータのうち構成情報に関するデータは構成管理部30に、障害情報に関するデータはイベントデータベース部26に、それぞれ適当な形式に変換して送る。

【0053】構成管理部30は、改善後の管理対象オブジェクトモデルと、これに関連した改善後のイベント伝播モデルとを保持する。管理対象オブジェクトモデルを図4に示す。図4には、図12に示した管理対象オブジェクトモデルのうち改善された部分のみを示す。図4に示される例では、Nodeクラス124およびLinkクラス126の間に、新たにToward-manager関係が定義されている。このToward-manager関係は、互いに接続されているNodeおよびLink間に設定されるものであり、あるオブジェクトに接続されているNodeまたはLinkのうち、NMSの方に接続されているNodeまたはLink、すなわちNMSにより近い方のNodeまたはLinkに対してのみこの関係を持たせるようにする。この関係により、あるオブジェクトから見てNMSの位置が判明する。この関係を本明細書では「方位関係」と呼ぶこととする。

【0054】具体的な例として図13に示されるネットワークについて考える。図15に示されるインスタンス

図において、EtherLink196は、EtherNode176およびEtherNode190に対してConnected-via の関係を有している。これは従来と同様である。本願発明ではさらに、EtherLink196は、NMS 170の側のEtherNode176に対してだけToward-managerという関係を持つ。すなわちEtherNode176から見てNMS 170はEtherNode176の方位にあると考えられるわけである。

【0055】そうして、このようにしてToward-managerの関係を定義した後に、EtherLink196の問題イベントDownが、このToward-managerの関係に沿って伝播すると定義する。このように症状イベントの伝播を定義するのが本願発明のイベント伝播ルールである。

【0056】改善されたイベント伝播ルールは次のとおりである。EtherLink がDownすればToward-manager関係に沿ってEtherNode がDownする。

【0057】EtherNode がDownすれば、Toward-manager関係に沿ってMacBridgeLink がCannotBridgingである。

【0058】なお、図2に示した管理対象オブジェクトモデルでは、Manager クラスとResourceクラスとの間でManages, Managed-by関係を持たせることができる。これは一見本願発明のToward-manager関係と類似しているかのように思われるが、このManages, Managed-by関係では問題を解決することはできない。

【0059】改善されたイベント伝播モデルを図5に図式的に示す。図5に示されるのは、図16に示される従来のイベント伝播モデルのうち本願発明により改善された部分のみである。図5を参照してわかるように、この第1の実施の形態では、EtherLink クラス164からEtherNode クラス158に対してToward-Manager関係74が、EtherNode クラス158からMacBridgeLink クラス166に対してToward-manager関係76がそれぞれ設定される。

【0060】再び図1を参照して、構成管理部30は、このように改善された後の管理対象オブジェクトモデルと、同じく改善された後のイベント伝播モデルとを保持する。なおこれらモデルは、図3に示すハードディスク54内のファイル(図示せず)から読込むこともできるし、ユーザインターフェイス部36を介してユーザが入力することも可能である。

【0061】構成管理部30はさらに、ネットワークインターフェイス部24から図13に示されるネットワーク例のような構成情報データを受取り、管理対象オブジェクトモデルに基づいて図15に示されるような管理対象オブジェクト構成情報を構築する。この管理対象オブジェクト構成情報は管理対象オブジェクトトリポジトリ32に保持される。構成管理部30は、管理対象オブジェクト構成情報の構築の際に、後に述べるような方法に従って自動的に各管理対象オブジェクト間に必要なToward-manager関係を設定する。

【0062】構成管理部30はその後、自己の保持する管

理対象オブジェクトモデルおよびイベント伝播モデル、ならびに管理対象オブジェクトリポジトリ32に保持された管理対象オブジェクト構成情報を用いて、症状イベントパターンマトリックスを生成する。構成管理部30は、症状イベントパターンマトリックスを障害管理部34に通知する。図6に、既に述べたケース1およびケース2について、この実施の形態に従って生成した症状イベントパターンマトリックスの例を示す。なお、問題イベントの指定は、イベント伝播モデル記述時にも可能であるし、ユーザインターフェイス部36からユーザが症状イベントパターンマトリックスの生成を要求する際に、イベント伝播モデルを参照しながら指定することも可能である。

【0063】一方イベントデータベース部26は、ネットワークインターフェイス部24から構成情報データおよび障害情報データをイベントとして受取り、イベントリポジトリ28に保持する。イベントデータベース部26はこれらイベントの中で、障害情報に関するデータ、特に障害の症状イベントを障害管理部34に通知する。

【0064】障害管理部34は、構成管理部30から通知された症状イベントパターンマトリックスを受取り保持する。障害管理部34はまた、イベントデータベース部26から障害の症状イベントの集合を受取る。この症状イベントの集合とはすなわち、既に図18に示される表2のように、実際に観測される症状イベントパターンである。障害管理部34は、構成管理部30から受取った症状イベントパターンマトリックスの症状イベントパターンと、イベントデータベース部26から受取った実際に観測された症状イベントパターンとを比較して、障害の根本原因の問題を特定する。特定された問題はユーザインターフェイス部36に通知される。ユーザインターフェイス部36は、図2および図3に示されるディスプレイ42などによりこの障害の根本原因の問題を特定する情報をユーザに提示する。

【0065】図6に示す、改良後のイベント伝播モデルに基づく症状イベントパターンを、図18に示す、実際に観測される症状イベントパターンと比較されたい。図6と図18とにおいて、同一のケースに対しては実際に観測される症状イベントパターンと改良後のイベント伝播モデルに基づく症状イベントパターンとが一致している。具体的には、ケース1（実際）とケース1（改善後の伝播モデル）との症状パターンの差は0であり、ケース1（実際）とケース2（改善後の伝播モデル）との症状パターンの差は5である。ケース2（実際）とケース2（改善後の伝播モデル）との症状パターンの差は0であり、ケース2（実際）とケース1（改善後の伝播モデル）との症状パターンの差は5である。すなわち本願発明が採用したモデル化はほぼ理想的なものであると考えられる。

【0066】ところで、Toward-Manager関係は、次に述

べるようにして構成管理部30が自動的に設定することができる。従来の管理対象オブジェクトモデルに基づく管理対象オブジェクトおよび関係が既に得られているものと仮定する。以下の説明は図15に示されているようなネットワークを例として示す。

【0067】NMSを表わすオブジェクト170から、クラス間に定義されている関係に沿って管理対象オブジェクトを辿っていく。図15に示される例では、TcpNode172, IpNode174, EtherNode176, EtherLink196という順序でオブジェクトを辿る。EtherLink196では、Connected-to関係のあるEtherNode176としてEtherNode190およびToward-managerがある。しかしこの場合、NMSのオブジェクト170から順にオブジェクトを辿ってきているので、直前のオブジェクトはEtherNode176であったことがわかっている。すなわち、EtherLink196から見てNMS側のオブジェクトはEtherNode176である。したがってEtherLink196のToward-manager関係をEtherNode176に対してのみ設定する。以下これを繰返すことにより、自動的に適切な管理対象オブジェクトにToward-managerを設定することができる。

【0068】図7に、上記した処理を実現するためのプログラムを疑似コード形式で示し、そのフローチャートを図8に示す。図8を参照してこの処理の概略を説明する。以下の説明ではグラフ理論の用語を適宜用いる。以下の説明で用いる記号を次のように定義する。

【0069】G：ネットワークを表現するグラフ。たとえば図15では、各オブジェクトをグラフの頂点とし、オブジェクト間の結合をエッジとして定義されるグラフを指す。

【0070】u, v：グラフの頂点。たとえば図15では、オブジェクト170、172、174等の個々の管理対象オブジェクトを示す。

【0071】(u, v)：グラフ中の頂点uおよびv間のエッジ。たとえば図15では、オブジェクト170および172の間の関係、オブジェクト172および200の間の関係を表わす。

【0072】Q：以下の処理で使用するキューで、「頂点」を含む。図8を参照して、まずQに処理を開始する頂点であるManager170を入れる(80)。続いてQが空かどうかを判定する(82)。空であれば処理終了である。空でなければ先頭頂点uを削除する(84)。その後先頭頂点uから出るすべてのエッジ(u, v)について以下の処理を完了したかどうかを判定する(86)。すべてのエッジ(u, v)に対して処理が終了していたら頂点uを次の頂点に進め(98)処理をステップ82に戻す。

【0073】ステップ86でuからのすべてのエッジ(u, v)に対する処理が終了していない場合、エッジ(u, v)がConnected-to関係であるかどうかを判定する(88)。判定結果がNOであれば制御はステップ9

4に進む。判定結果がYESであれば続いて、Toward-manager-relation (u) がvでないかどうかの判定を行なう(90)。vでない場合にはToward-manager-relation (v) =uとする(92)。そしてvがQに存在しないかどうかについての判定を行なう(94)。vがQに存在する場合には制御はステップ86に進む。vがQに存在しない場合にはQの末尾にvを挿入して(96)制御をステップ86に戻す。

【0074】こうした制御構造を有するソフトウェアによりToward-manager関係を自動的に設定することができる。

【0075】一方、ネットワークの接続関係に変更があった場合を想定する。たとえばNMSが図13に示される第1のホスト130から第2のホスト132に移ったりする場合である。この場合、そうした情報をユーザインターフェイス部36から構成管理部30に与える。構成管理部30は、この情報が与えられると、管理対象オブジェクトに設定されているToward-manager関係をすべて一旦削除する。その後上記したToward-manager関係の自動設定を行なう。この方法により、ホストの移動のみならず、ネットワークの接続関係に変更があった場合にも自動的にToward-manager関係の自動設定変更を行なうことができ、常に正確に障害の根本原因の問題の特定を行なうことができる。

【0076】実施の形態2

実施の形態1では、互いに接続しているNodeおよびLink間で、NMSの方に接続されているオブジェクトに対してのみToward-managerという関係を持たせた。この関係の定義は、NMSに対する方位関係を定義できればよいので、このToward-manager関係には限定されない。たとえばNMSの反対側に接続されているオブジェクトに対してのみたとえばAway-from-managerという関係を持たせるようにしてもよい。その例を図9に示す。図9に示される関係100がAway-from-managerである。そして症状伝播モデルとして、たとえばEtherLinkがDownしたときに、Away-from-manager関係に沿ってCannotObserveというような観測されないイベントを伝播させ、そのイベントが、NMSから離れた管理対象オブジェクトのヘルスチェックタイムアウトを引き起こすといったモデル化を行なうことも可能である。そのモデル化の例を図式的に図10に示す。図10において、EtherLinkクラスとLogicalServiceクラスとの間にAway-from-manager関係102が設定される。同じくEtherNodeクラスとLogicalServiceクラスとの間にもAway-from-manager関係104が設定される。そしてこの関係102に沿ってDownが伝播してCannotObserveが生じ、それがさらに関係104により伝播してHealthCheckTimeoutを引き起こす。このモデル化によっても実施の形態1と全く同じ効果を期待することができる。

【0077】実施の形態3

なお実施の形態1および2では、図1に示すようにネットワークインターフェイス部24が直接管理対象ネットワーク22に接続されている。これに対して図9に示すように、ネットワークインターフェイス部24が既存のネットワーク管理システム110を介して管理対象ネットワーク22に接続されてもよい。この場合ネットワークインターフェイス部24は、既存のネットワーク管理システム110の保持する構成情報データおよび障害情報データを取得し、適当な形式に変換してそれぞれ構成管理部30およびイベントデータベース部26に与えることになる。こうした構成でも上記した効果を得ることができる。

【図面の簡単な説明】

【図1】本願発明の実施の形態1に係るネットワーク管理システムのブロック図である。

【図2】本願発明に係るネットワーク管理システムを実現するためのコンピュータの外観図である。

【図3】図2に示されるコンピュータのブロック図である。

【図4】管理対象オブジェクトモデルのうち改善部分のみを示す図である。

【図5】イベント伝播モデルのうち、改善された部分のみを示す図である。

【図6】実施の形態1において採用されたイベント伝播モデルに基づく症状イベントパターンを表形式で示す図である。

【図7】Toward-manager関係を自動的に設定するプログラムの疑似コードを示す図である。

【図8】図7に示される疑似コードのフローチャートである。

【図9】実施の形態2で採用される管理対象オブジェクトモデルを示す図である。

【図10】実施の形態2で用いられるイベント伝播モデルを示す図である。

【図11】実施の形態3のネットワーク管理システムのブロック図である。

【図12】従来の技術で使用される管理対象オブジェクトモデルのクラス図である。

【図13】簡単なネットワークを模式的に示す図である。

【図14】図13に示されるネットワークの管理対象オブジェクトモデルのクラス図である。

【図15】図13に示されるネットワークの管理対象オブジェクトモデルを示すインスタンス図である。

【図16】従来のイベント伝播モデルのクラス図である。

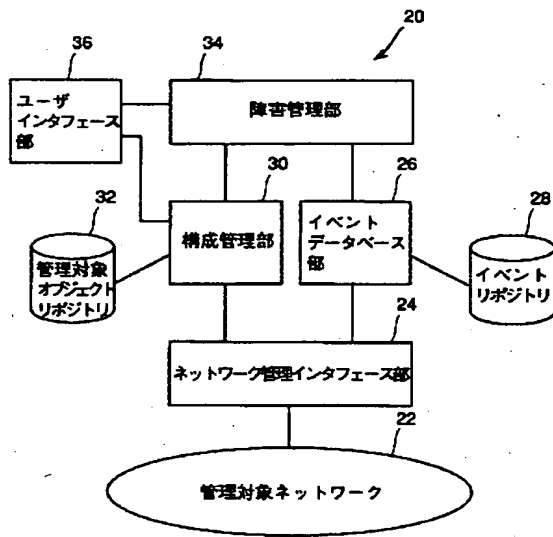
【図17】従来のイベント伝播モデルに基づく症状イベントパターンを表形式で示す図である。

【図18】NMSにより実際に観測される症状イベントパターンを表形式で示す図である。

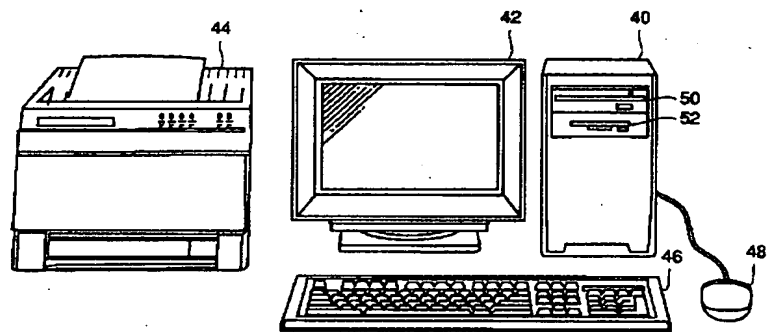
- 17
 20 ネットワーク管理システム
 22 管理対象ネットワーク
 24 ネットワークインタフェース部
 26 イベントデータベース部
 28 イベントリポジトリ
 30 構成管理部
 32 管理対象オブジェクトリポジトリ
 34 障害管理部
 36 ユーザインタフェース部
 40 コンピュータ本体
 42 ディスプレイ
 44 プリンタ

- 46 キーボード
 48 マウス
 50 CD-ROMドライブ
 52 FDドライブ
 54 ハードディスク
 56 CPU
 58 ROM
 60 RAM
 62 CD-ROM
 64 フレキシブルディスク
 70 Toward-manager関係
 100 Away-from-manager関係

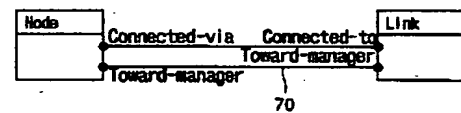
【図1】



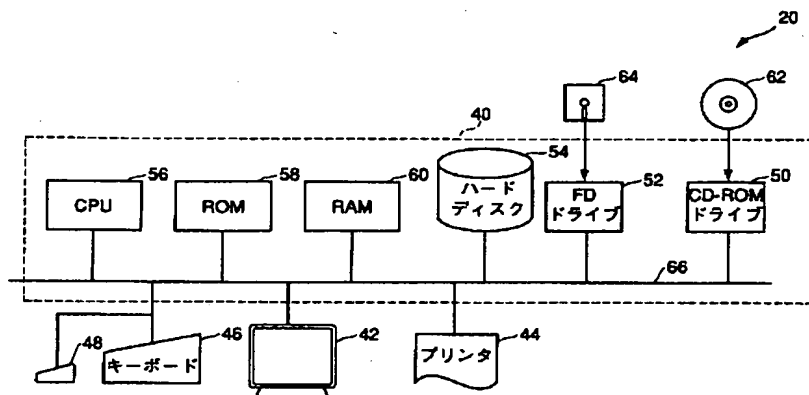
【図2】



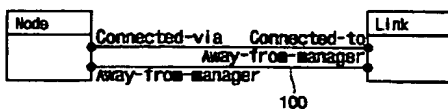
【図4】



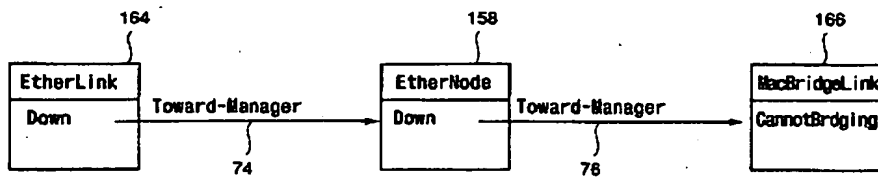
【図3】



【図9】



【図5】



【図6】

改善後のイベント伝播モデルに基づく症状イベントパターン

問題イベント 症状イベント	ケース1： EtherLink1.Down	ケース2： EtherLink1.Down
EtherNode1.Down	1	
TcpNode1.Disconnect	1	1
EtherNode2.Down		
TcpNode2.Disconnect		
EtherNode3.Down		
EtherNode4.Down		1
MacBridgeNode1.CannotBridging		1
EtherLink1.Down	1	
EtherLink2.Down		1
TcpLink1.NotResponding	1	1

【図17】

イベント伝播モデルに基づく症状イベントパターン

問題イベント 症状イベント	ケース1： EtherLink1.Down	ケース2： EtherLink1.Down
EtherNode1.Down	1	
TcpNode1.Disconnect	1	1
EtherNode2.Down		1
TcpNode2.Disconnect	1	1
EtherNode3.Down	1	
EtherNode4.Down		1
MacBridgeNode1.CannotBridging	1	1
EtherLink1.Down	1	
EtherLink2.Down		1
TcpLink1.NotResponding	1	1

【図7】

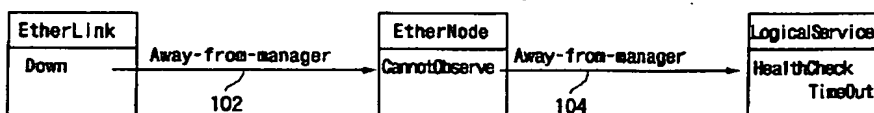
Qに、処理を開始する頂点であるMANAGER1
を入れる

```

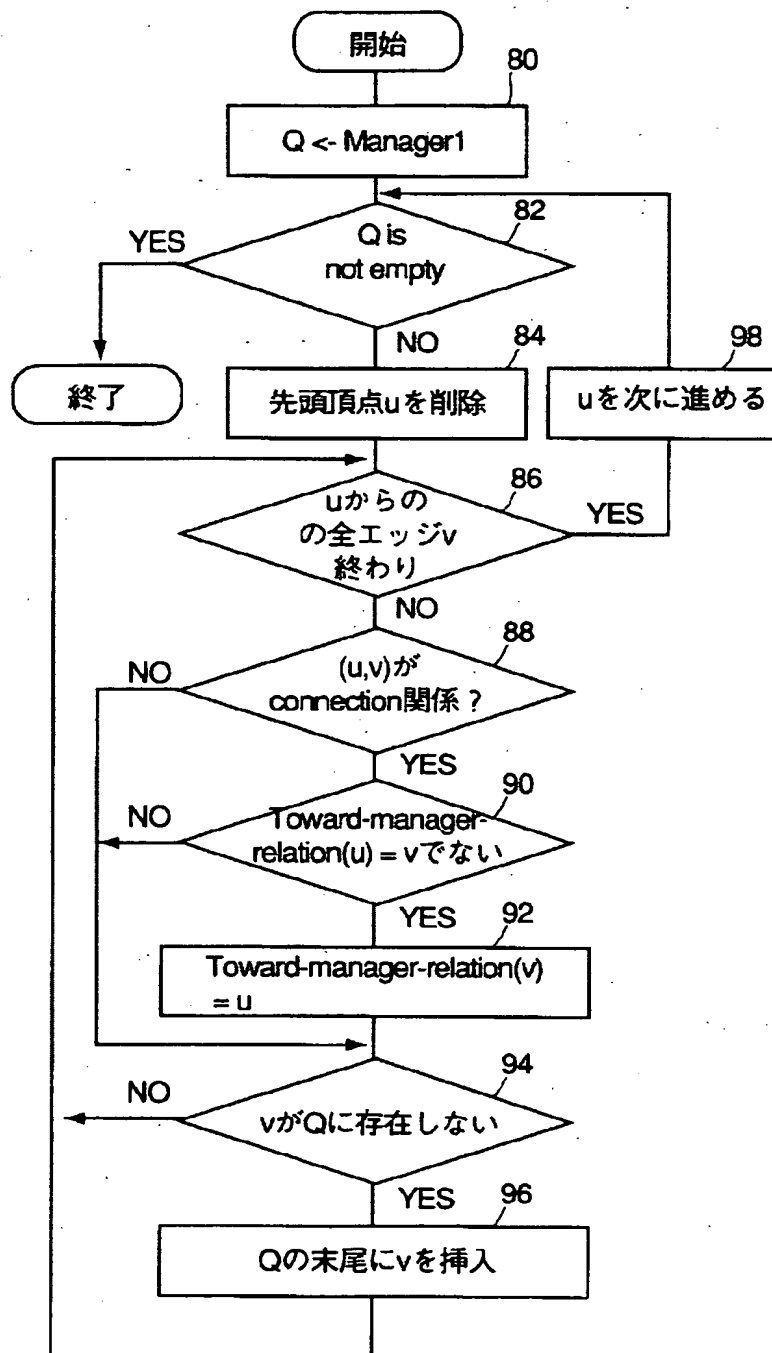
While ( Q is not empty ) {
  delete Q's head vertex u;
  for ( each edge(u,v) starting at u ) {
    if ( relation name of edge(u, v)がConnection関係である )
      if ( Toward-manager-relation(u)がvでない )
        Toward-manager-relation(v) = u;
    if ( vがQに存在しない )
      insert v at the tail of Q;
  }
}

```

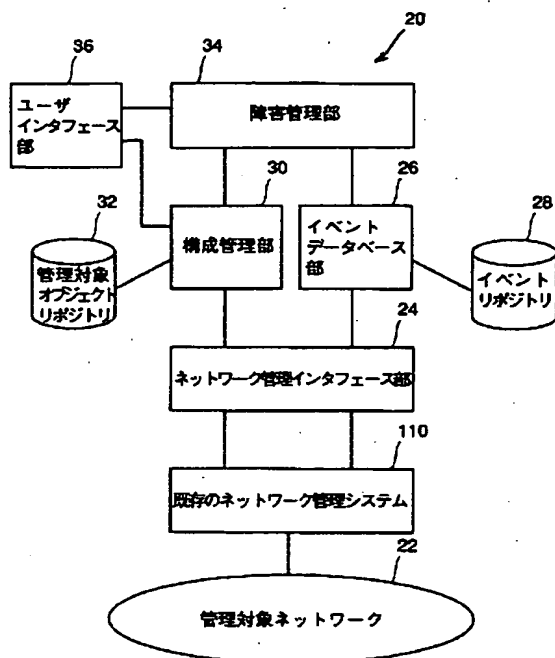
【図10】



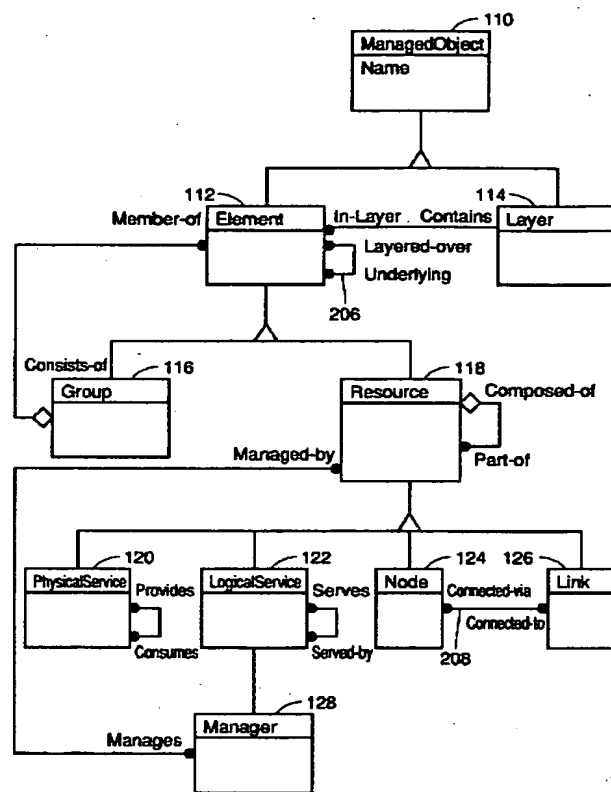
【図8】



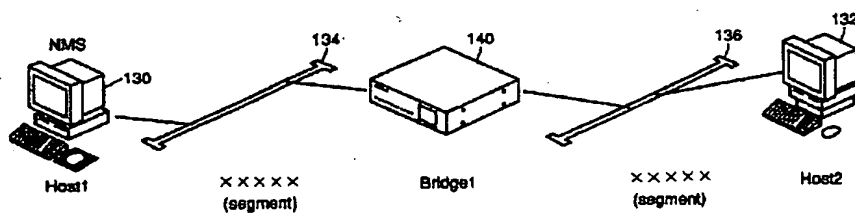
【図11】



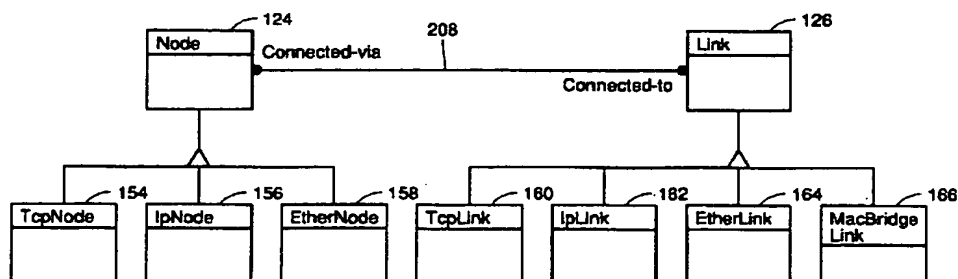
【図12】



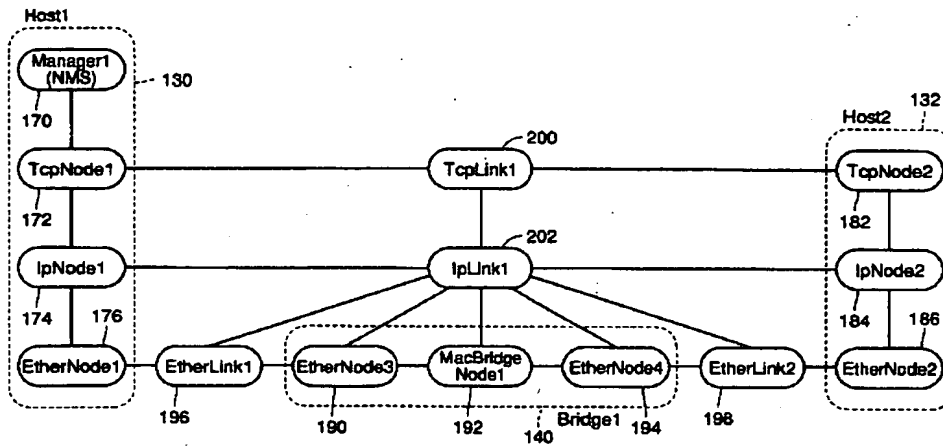
【図13】



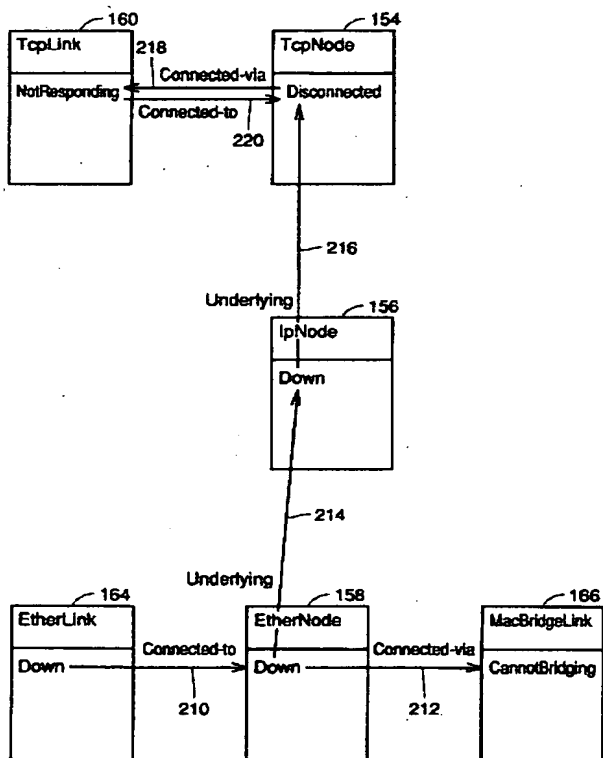
【図14】



【図15】



【図16】



【図18】

NMSにより実際に観測される症状イベントパターン

問題イベント 症状イベント	ケース1： EtherLink1.Down	ケース2： EtherLink1.Down
EtherNode1.Down	1	
TcpNode1.Disconnect	1	1
EtherNode2.Down		
TcpNode2.Disconnect		
EtherNode3.Down		
EtherNode4.Down		1
MacBridgeNode1.CannotBridging		1
EtherLink1.Down	1	
EtherLink2.Down		1
TcpLink1.NotResponding	1	1